UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/723,863	11/26/2003	Jason P. Chalecki	MS305499.01	9696

69316 7590 09/19/2007	EXAMINER	
MICROSOFT CORPORATION	TRAN, QUOC A	
ONE MICROSOFT WAY		
REDMOND, WA 98052	ART UNIT	PAPER NUMBER
	2176	

MAIL DATE	DELIVERY MODE
09/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/723,863	CHALECKI ET AL.
	Examiner	Art Unit	
	Tran A. Quoc	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>06/28/2007</u>.

2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-45</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-18, 20-23. 25-27, and 29-44</u> is/are rejected.

7) ☒ Claim(s) <u>19, 21, 24, 26, 28, 30, 43, and 45</u> is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on <u>28 June 2007</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>06/28/07; 08/13/2007; 9/14/07.</u>

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

This is a **Final rejection** in response to amendment filed 06/28/2007.

Claims 1-45 are pending. Claims 1, 21, 26, 33, 37, and 41 are independent

claims. Effective filing date is 11/26/2003, which claims CIP of 10/610, 504 filed 06-30-

2003.

Object to claims 14, and 43 of the previous rejection mailed 02/08/2007 are

withdrawn due to applicants' amendments.

Claims Objection

Regarding claim(s) 21, 26, 30 and 43 the phrase **"exceeds a threshold"**, and

"the likelihood", and **"a combination of forgoing"** indefinite (Applicant's "In The

Claims" pages 12, 14, 16 and 23), renders the claimed indefinite, because Applicant's

invention specification merely discloses " script-based custom the validation rule, and

real-time validation rule.... nodes governing by other nodes. Thus the real-time

validation tool can validate data from a data-entry filed" (Applicant's invention

specification para [0122]), and the validation tool item 136 can build validation rules

from schema contain logic that governs a structure data file. This logic sets forth the

bounds of what are the nodes..." (Applicant's invention specification para 0124). It is

unclear what Applicant's intended the metes and bounds of the claims are, thereby

rendering the scope of the claim(s) unascertainable (an open-ended ranges). Regarding

claim 13 the phase "one or ore" is assumed to be a typo error of "one or more".

Allowable Subject Matter

Claims 19, 24, 28, and 45 objected to as being dependent upon a rejected base

claim, but would be allowable if rewritten in independent form including all of the

limitations of the base claim and any intervening claims.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

Claims 1-18, 20-23, 25-27, and 29-44 are rejected under 35 U.S.C. 103(a) as

being unpatentable over Altova **et al. "www. XML Spy.com spy 4 you" Published**

05/24/2002 (hereinafter "Altova"), in view of XForm 1.0 by W3C Published

07/16/2001 (hereinafter "XForm 1.0").

Regarding independent claims 1, Altova teaches:

> **A method comprising: receiving an instruction to open an**
>
> **eXtensible Markup Language (XML) document;**

(See Altova at Page 11, discloses instruction to open an XML document.

opening the XML document with the solution, wherein: the solution

includes an extensible stylesheet language (XSLT) presentation

application and a XML schema; the XML document can be inferred

from the XML schema;

executing the XSLT presentation application to render a Hypertext
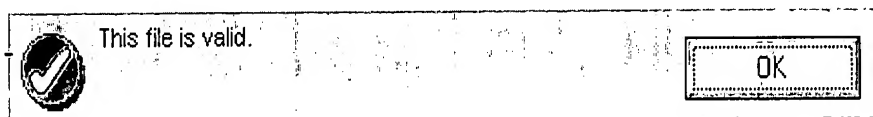
Markup Language (HTML).

(See Altova at Page 10, discloses XML Spy user interface.

Clicking the "Display as Table" icon, [icon] activates the Database/Table view.
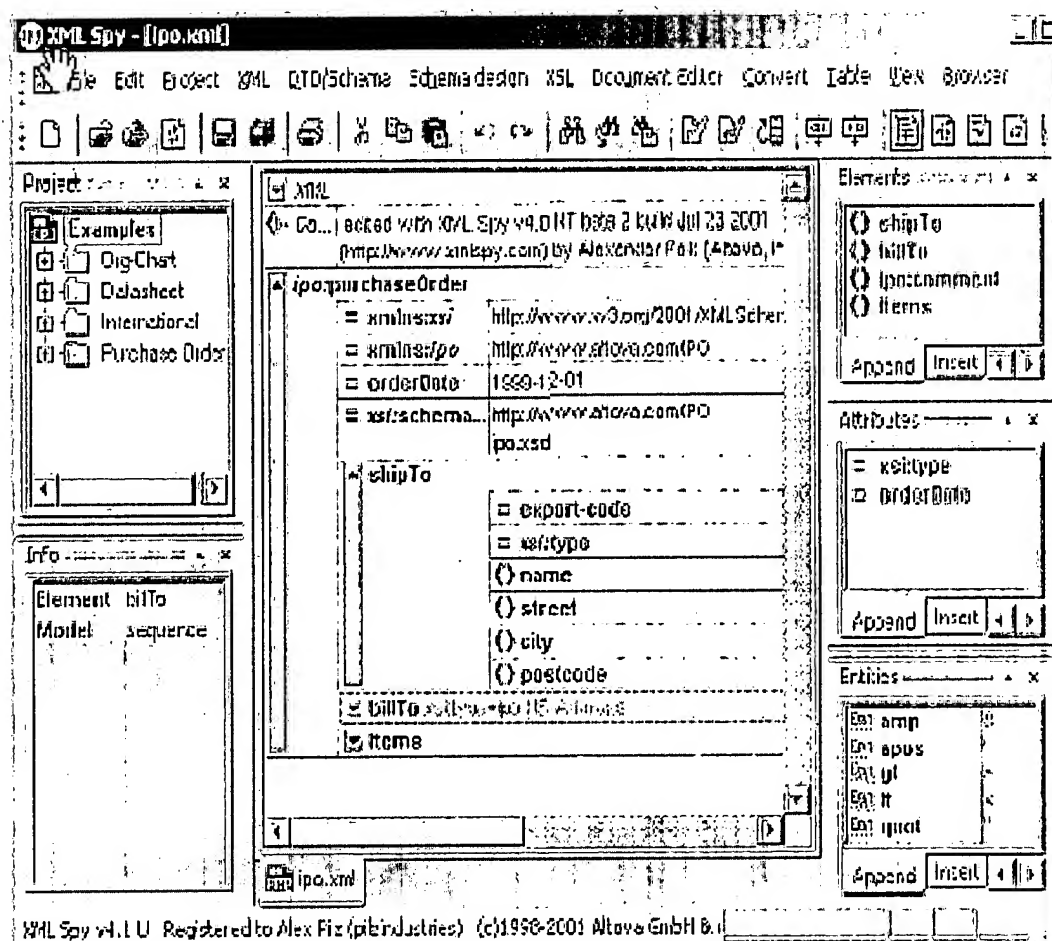
Also, see Altova Page 12, discloses the XML Spy Document Framework and XML Spy IDE. XML Spy Document Framework consists of XSLT Designer and XML Spy Document Editor. Help on each member of the product family is available using the Help menu option.

Also, see XML Spy page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new W3C XML Schema Definition Language (XSD),

This file is valid.		OK

If an error is encountered during the validation, the source of the problem is highlighted and a corresponding error message is shown:

This file is not valid.		Revalidate
Mandatory element 'content' expected in place of 'table'		
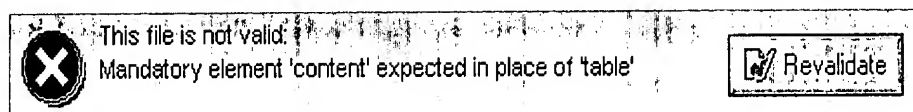
receiving, through the data-entry fields, data input by a user; validating the data input by the user; if the act of validating determines that the data input by the user is invalid, outputting indicia informing the user that the data input is invalid.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to interact with the user, where the value return true or false.

Also, see Altova page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification.)

and portions of the XML document are logically coupled with fragments of the XML schema;

(See Altova page 235, discloses XML Spy allows you to import hierarchical database data using the ADO data shaping function, this allows you to re-import specific parts of external databases.

Also, see Altova page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification.)

executing the XSLT presentation application to render a Hypertext Markup Language (HTML) electronic form containing data-entry fields associated with the coupled portions;

(See Altova page 44, depicts in the diagram below, the first page of the schema

documentation in HTML form. If components from other schemas have been included,

then those schemas are also documented.

Also see Altova Page 69, discloses the process of transforming XML to HTML.)

Schema **AddressLast.xsd**

schema location: **C:\Program Files\Altova\XML Spy Suite\Examples\Tutorial\AddressLast.xsd**
targetNamespace: **http://my-company.com/namespace**

Elements	Complex types	Simple types
Company	**AddressType**	**US-State**
Person	**UK-Address**	
	US-Address	

element **Company**

diagram	
namespace	http://my-company.com/namespace
children	**Address Person**
annotation	documentation Root element

source	`<xs:element name="Company">`
	`  <xs:annotation>`
	`    <xs:documentation>Root element</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="Address" type="AddressType"/>`
	`      <xs:element ref="Person" maxOccurs="unbounded"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

<u>In addition, Altova does not teach, but XForm 1.0 teaches:</u>

searching the XML document to locate a processing instruction (PI)

containing a href attribute that points to a URL; discovering a solution

using the URL in the processing instruction (PI).

(See XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model> Element,

discloses the namespace prefixes "xform:" and "xsd:" are used to denote

the XForms and XML Schema namespaces respectively. This is by convention only;

any namespace prefix may be used in practice. For example:

<foo href="http://www.example.com/XForms"/>

Also see XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model>

Element, discloses

```
<xform xmlns="http://www.w3.org/2001/02/xforms" id="Person">

    <model href="Schema-Questionnaire.xform" />

    <instance href="URL-to-retrieve-defaults" />

</xform>
```

and also,

```
<model

        id = xsd:ID

        href = xsd:uriReference

        >

        <!-- Content: (xsd:schema | simple) -->

</model>.
```

It would have been obvious to one of ordinary skill in the art at the time of the

invention to modify Altova XML Spy validating process, where validates an XML

document against the rules set forth in its Schema or Document Type Definition (DTD)

or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification of the schema documentation in HTML form, to include the

searching the XML document feature to locate a processing instruction (PI) containing a

href attribute that points to a URL; discovering a solution using the URL in the

processing instruction (PI) of XForm 1.0 to improve the validating process of Altova

XML Spy for the predictable result of better Web forms with richer interactions, for

online interaction between a XForms Processor and a remote entity as taught by XForm

1.0 at section 1.1 the Background.)

Regarding independent claims 21,

the rejection of claim 1 is fully incorporated.

In addition, Altova teaches:

examining the name of the PI to assess the likelihood that the

PI includes a solution identifier for the solution; and when the

likelihood exceeds a threshold, discovering a solution using the

name in the PI;

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to

interact with the user, where the value return true or false.

Also, see Altova page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification. It is noted the claimed **the likelihood** is the value return true or false as taught by Altova XML Spy.)

Regarding independent claims 26, the rejection of claims 1 and 21 are fully incorporated, and are similarly rejected under the same rationale.

Regarding independent claims 33, 37, the rejection of claim 1 is fully incorporated, and is similarly rejected under the same rationale.

Regarding independent claims 41,

is directed a computer-readable medium to perform the method of claim 1 which cites above, and is similarly rejected under the same rationale.

Claims 2, 22, 29, 34, 38 and 42,

wherein one or more of the receiving, the searching, the examining, the discovering, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

(See Altova page 44, depicts in the diagram below, the first page of the schema

documentation in HTML form. If components from other schemas have been included,

then those schemas are also documented.

Also see Altova Page 69, discloses the process of transforming XML to HTML.)

Schema **AddressLast.xsd**

schema location: **C:\Program Files\Altova\XML Spy
                     Suite\Examples\Tutorial\AddressLast.xsd**
targetNamespace: **http://my-company.com/namespace**

Elements   Complex types   Simple types
Company AddressType US-State
Person   UK-Address
            US-Address

element **Company**

diagram	
namespace	http://my-company.com/namespace
children	**Address Person**
annotation	documentation Root element

source	`<xs:element name="Company">`
	` <xs:annotation>`
	`  <xs:documentation>Root element</xs:documentation>`
	` </xs:annotation>`
	` <xs:complexType>`
	`  <xs:sequence>`
	`   <xs:element name="Address" type="AddressType"/>`
	`   <xs:element ref="Person" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	` </xs:complexType>`
	`</xs:element>`

Also, see Altova Page 12, discloses the XML Spy Document Framework and

XML Spy IDE. XML Spy Document Framework consists of XSLT Designer and XML

Spy Document Editor. Help on each member of the product family is available using the

Help menu option.

Claim 3, Altova teaches:

wherein: the executing the XSLT presentation application comprises

applying an XSLT stylesheet to the XML document to create the HTML

electronic form; and the HTML electronic form includes a plurality of HTML

elements corresponding to the data-entry fields.

(See XML Spy pages 68-69, teaching XSL Transformation XSL | Assign XSL, assigns an

XSL file to an XML document, transforms the XML document(s) into the files

specified by the XSL Transformation document, in this case into an HTML form.

See also XML Spy Page 71, teaching XML Spy product family includes XSLT

Designer, which automates the generation of XSLT stylesheets for XML to HTML

transformation.)

Claim 4,

In addition, Altova does not teach, but XForm 1.0 teaches:

the searching the XML document further comprises finding the first

PI in the XML document; and the first PI in the XML document includes the

URL.

(See XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model> Element,

discloses the namespace prefixes "xform:" and "xsd:" are used to denote

the XForms and XML Schema namespaces respectively. This is by convention only;

any namespace prefix may be used in practice. For example:

<foo href="http://www.example.com/XForms"/>

Also see XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model>

Element, discloses

```
<xform xmlns="http://www.w3.org/2001/02/xforms" id="Person">

    <model href="Schema-Questionnaire.xform" />

    <instance href="URL-to-retrieve-defaults" />

</xform>
```

and also,

```
<model

        id = xsd:ID

        href = xsd:uriReference

        >

        <!-- Content: (xsd:schema | simple) -->

</model>.
```

It would have been obvious to one of ordinary skill in the art at the time of the

invention to modify Altova XML Spy's validating process, where validates an XML

document against the rules set forth in its Schema or Document Type Definition (DTD)

or it can validate any XML Schema or DTD against the rules set forth in the
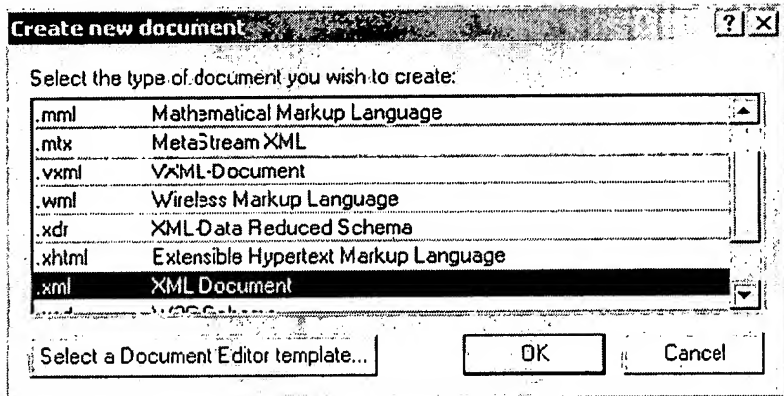
corresponding specification of the schema documentation in HTML form, to include the

searching the XML document feature to locate a processing instruction (PI) containing a

href attribute that points to a URL; discovering a solution using the URL in the

processing instruction (PI) of XForm 1.0 to improve the validating process of Altova

XML Spy for the predictable result of better Web forms with richer interactions, for

online interaction between a XForms Processor and a remote entity as taught by XForm
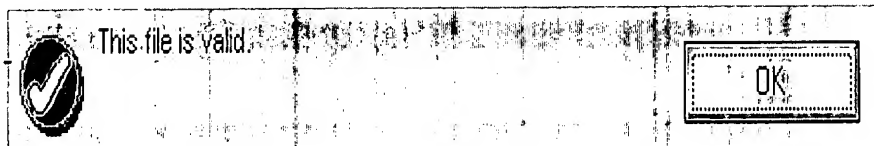
1.0 at section 1.1 the Background.

Claim 5, Altova teaches:

wherein the solution further comprises a manifest of all files that can

be used for: representing the XML document in the HTML electronic form;

allowing the user to input data into the data-entry fields; and the act of

validating the data that the user inputs into the data-entry fields.

(See Altova page 446, teaching Opening a document template *.sps, where XML Spy

Document Editor enables you to edit XML documents based on templates created in

XSLT Designer! You cannot create a new template using Document Editor; this must be

done in XSLT Designer. Using the broadest reasonable interpretation, the Examiner

reads the claimed **a manifest of all files** as equivalent to *.sps as shows in Fig below:

Create new document [?] [X]

Select the type of document you wish to create:

.mml	Mathematical Markup Language
.mtx	MetaStream XML
.vxml	VXML-Document
.wml	Wireless Markup Language
.xdr	XML-Data Reduced Schema
.xhtml	Extensible Hypertext Markup Language
.xml	XML Document

Select a Document Editor template...   OK   Cancel

Also, see Altova page 180, discloses the validating process, where validates an

XML document against the rules set forth in its Schema or Document Type Definition

(DTD) or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new

W3C XML Schema Definition Language (XSD),

This file is valid.   OK

If an error is encountered during the validation, the source of the problem is highlighted and a
corresponding error message is shown:

This file is not valid.
Mandatory element 'content' expected in place of 'table'   Revalidate

Claim 6, XML Spy teaches:

> **wherein the coupled portions contain information setting forth all**
>
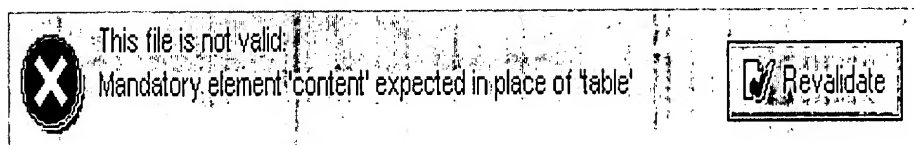> **possible XML documents for the coupled portions.**

(See XML Spy page 446, teaching Opening a document template *.sps, where XML Spy

Document Editor enables you to edit XML documents based on templates created in

XSLT Designer! You cannot create a new template using Document Editor; this must be

done in XSLT Designer. Using the broadest reasonable interpretation, the Examiner

reads the claimed **all possible XML document** as equivalent to *.sps as taught by

Altova XML Spy.)

Regarding claims 7, 23, 27, 35, 39 and 44, Altova teaches:

> **the data-entry fields of the HTML electronic form map to a**
>
> **corresponding plurality of nodes of the XML document; and the data input**
>
> **is input for storage in a corresponding said node in the XML document and**
>
> **further comprising outputting data in XML for viewing by the user in the**
>
> **HTML electronic form through the data-entry fields via the mapping of the**
>
> **data- entry fields from corresponding said nodes of the XML document.**

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to

interact with the user, where the value return true or false.

Also, see Altova page 180, discloses the validating process, where validates an

XML document against the rules set forth in its Schema or Document Type Definition

(DTD) or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification.

See also, Altova page 235, discloses XML Spy allows you to import hierarchical

database data using the ADO data shaping function, this allows you to re-import specific

parts of external databases.

See also, Altova page 44, depicts in the diagram below, the first page of the

schema documentation in HTML form. If components from other schemas have been

included, then those schemas are also documented.

Also see Altova Page 69, discloses the process of transforming XML to HTML.)

Claim 8, Altova teaches:

> **the XML schema includes a logic application and the act of validating**
>
> **executes the logic application.**

(See XML Spy page 180, discloses the validating process, where validates an XML

document against the rules set forth in its Schema or Document Type Definition (DTD)

or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new·

W3C XML Schema Definition Language (XSD),) ·

Claim 9, Altova teaches:

> **the act of validating is performed on the data input from the user into**
>
> **each said data-entry field with one or more of a plurality of validation rules**

> **for; the logic application comprises a plurality of the validation rules for: a corresponding plurality of the nodes in the XML document; and a corresponding plurality of the data-entry fields; the act of validating uses each said validation rule to: determine if the data received by input from the user into a corresponding said data-entry field is valid or invalid; and require the user to correct any data input into the corresponding said data-entry field that the validation determines to be invalid.**

(See Altova page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new W3C XML Schema Definition Language (XSD),

See also, Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to interact with the user, where the value return true or false.

Also, see XML Spy page 180, discloses the validating process, where validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification.

See also, Altova page 235, discloses XML Spy allows you to import hierarchical database data using the ADO data shaping function, this allows you to re-import specific parts of external databases.

See also, Altova page 44, depicts in the diagram below, the first page of the

schema documentation in HTML form. If components from other schemas have been

included, then those schemas are also documented.

Also see Altova Page 69, discloses the process of transforming XML to HTML.)

Claim 10, Altova teaches:

validation rule has an identity that is selected from the group

consisting of: the identity is based on a part of a schema governing a

corresponding said node; the identity is written in script and associated

with a corresponding said node; and the identity is written in a declarative

syntax and associated with a corresponding said node.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to

interact with the user, where the value return true or false.

See also Altova page 180, discloses the validating process, where validates an

XML document against the rules set forth in its Schema or Document Type Definition

(DTD) or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new
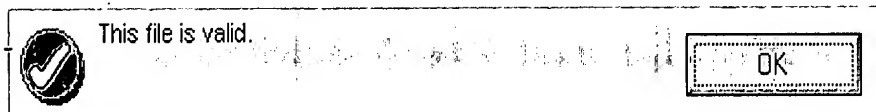
W3C XML Schema Definition Language (XSD),)

Claim 11, Altova teaches:

each said validation rule includes an alert area display and the

**indicia comprises the alert area display; and the act of validating further**

comprises using one said validation rule to determine that the data

received by input from the user into a corresponding said data-entry field is

invalid; and the act of outputting indicia outputs the corresponding alert

area display so as to be associated with the corresponding said data-entry

field.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to

interact with the user, where the value return true or false.

See also, Altova page 180, discloses the validating process, where validates an

XML document against the rules set forth in its Schema or Document Type Definition

(DTD) or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new

W3C XML Schema Definition Language (XSD),

This file is valid.   OK

If an error is encountered during the validation, the source of the problem is highlighted and a
corresponding error message is shown:

This file is not valid:
Mandatory element 'content' expected in place of 'table'   Revalidate
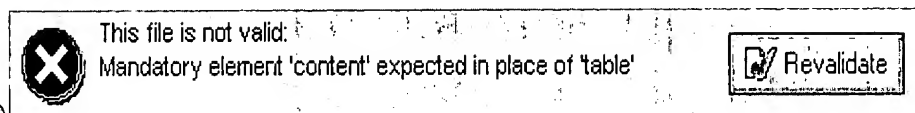
Claim 12, Altova teaches:

> wherein when the alert area display is output, the output includes
> one or more characteristics selected from the group consisting of:
> graphics surrounding the corresponding said data-entry field; the alert area
> display surrounds the corresponding said data-entry field; the alert area
> display includes graphics containing a red, dashed-lined box; the alert area
> display includes graphics highlighting the data in the corresponding said
> data-entry field; the alert area display surrounds the corresponding said
> data-entry field and includes the graphics containing a squiggly line
> beneath the data in the corresponding said data-entry field; the alert area
> display includes text containing information about the invalid data in the
> corresponding said data-entry field; the alert area display includes text
> containing corresponding said data-entry field; and the alert area display
> includes a pop-up window.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to interact with the user, where the value return true or false.

Also see Altova Page 55-56, discloses error massage of validating against input data field graphic windows as shows bellows:
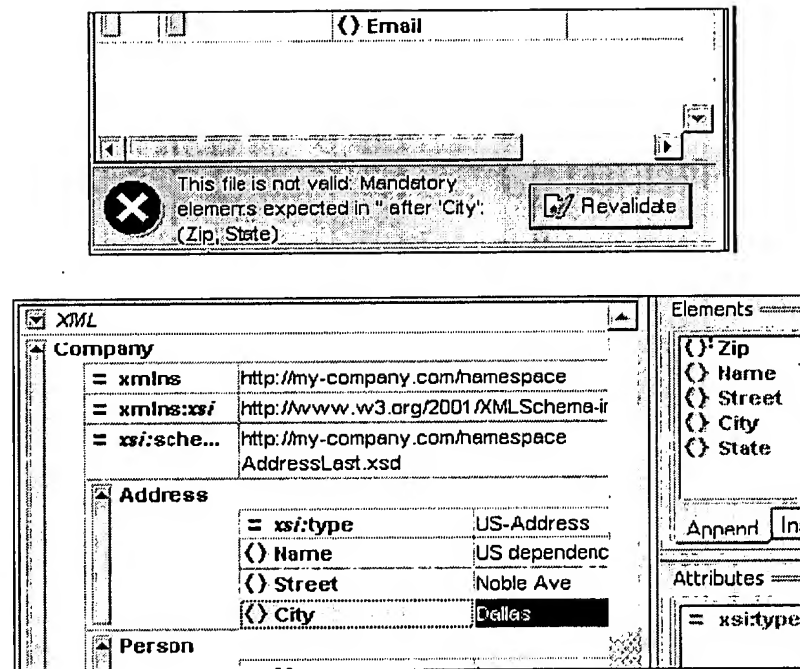
Using the broadest reasonable interpretation, the Examiner equates the claimed as

equivalent to forms dialog boxes enable scripts and Altova XML Spy to interact with the

user, where the value return true or false and its GUI as taught by Altova XML Spy.)

Claim 13, Altova teaches:

wherein each said node has one or ore of the validation rules

associated therewith.

(See Altova page 180, discloses the validating process, where validates an XML

document against the rules set forth in its Schema or Document Type Definition (DTD)

or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new

W3C XML Schema Definition Language (XSD),)

Claim 14, Altova teaches:

> wherein one said validation rule includes a requirement that is
> selected from the group consisting of: the data received by input from the
> user into a corresponding said data-entry field is to be numerical;
>
> the data received by input from the user into a corresponding said
> data-entry field is to be textual; and
>
> the data received by input from the user into a corresponding said
> data-entry field that references another said node in the data file.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to
interact with the user, where the value return true or false.

See also, Altova page 180, discloses the validating process, where validates an
XML document against the rules set forth in its Schema or Document Type Definition
(DTD) or it can validate any XML Schema or DTD against the rules set forth in the
corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new
W3C XML Schema Definition Language (XSD),)

Claim 15, Altova teaches:

> wherein the plurality of the validation rules re associated by
> mapping to the corresponding plurality of the nodes in the XML document.

(See Altova page 180, discloses the validating process, where validates an XML
document against the rules set forth in its Schema or Document Type Definition (DTD)
or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification, such as DTD, DCD, XDR, XML-Data, BizTalk, and the new

W3C XML Schema Definition Language (XSD),)

Claim 16, Altova teaches:

>   **validation rule is associated by mapping to a corresponding said**
>
>   **data-entry field by use of an entity selected from the group consisting of an**
>
>   **XPath expression, a declarative syntax, and an entity that is script-based.**

(See Altova Page 176, teaching Evaluate Xpath, XML Spy includes an XPath Visualizer

that allows you to define/check XPath expressions in your XML documents and display

the results immediately.

>   See also, ALtova page 323, discloses forms dialog boxes enable scripts and XML

Spy to interact with the user, where the value return true or false.)

Claim 17, Altova teaches:

>   **wherein the script-based validation rule maps to a corresponding**
>
>   **said node with an entity selected from the group consisting of:**
>
>   **an XPath expression;**
>
>   **an event handler;**
>
>   **an event handier that determines when a real-time validation**
>
>   **tool uses the script-based validation rule;**

> an event handler that determines when a real-time validation tool uses the script-based validation rule before data received for the node is held by the data file; and

> an event handler that determines when a real-time validation tool uses the script-based validation rule after data received for the node is held by the data file.

(See Altova Page 176, teaching Evaluate Xpath, XML Spy includes an XPath Visualizer that allows you to define/check XPath expressions in your XML documents and display the results immediately.

See also, Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to interact with the user, where the value return true or false.)


Claim 18, Altova teaches:

> wherein each said validation rule includes an alert area display; and

> how the alert area display is to appear when output.

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to interact with the user, where the value return true or false.

Also see Altova Page 55-56, discloses error massage of validating against input data field graphic windows as shows bellows:
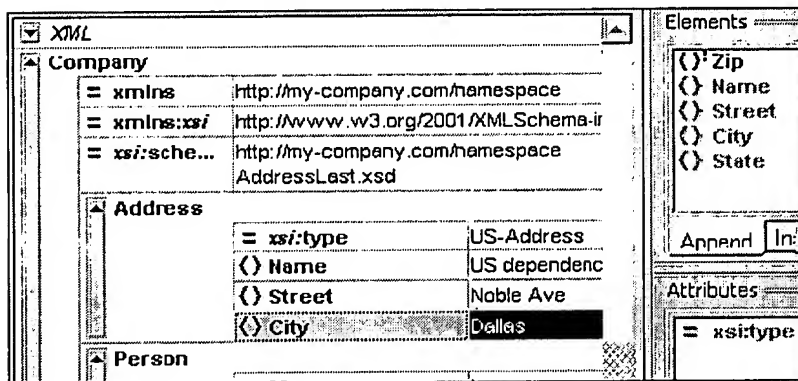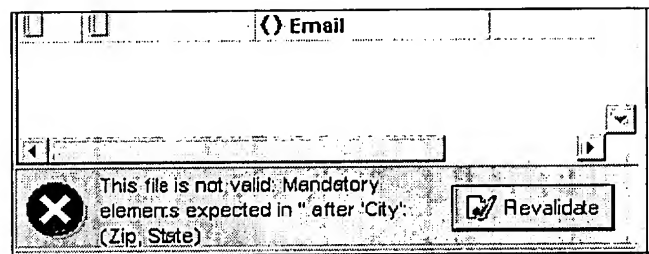
Claims 20, 25, 32, 36, 40,

the rejection of claims 1, 21, 26, 33, and 37 are fully incorporated.

In addition, a computer readable medium, when executed by a computer

embodied therein for performance the method of claims 1, 21, 26, 33, and 37 as

cite above.

(See Altova Page 6 bottom, teaching XML documents are per definition Unicode-based,

but may be stored on disk or transmitted over the network in various different

"encodings", such as ISO-8859-1 or UTF-8.)

Claim 30, Altova teaches:

the assessment of the likelihood exceeds the threshold,

(See Altova page 323, discloses forms dialog boxes enable scripts and XML Spy to

interact with the user, where the value return true or false.

Also see Altova Page 55-56, discloses error massage of validating against input

data field. In the broadest reasonable interpretation, Examiner interprets the claimed

the likelihood exceeds the threshold as equivalent to the value return true or false as

taught by Altova XML Spy.

In addition, Altova does not teach, but XForm 1.0 teaches:

when the PI is the first PI in the XML document that contains a URL;

the discovering a solution comprises using the first URL to discover the

solution.

(See XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model> Element,

discloses the namespace prefixes "xform:" and "xsd:" are used to denote

the XForms and XML Schema namespaces respectively. This is by convention only;

any namespace prefix may be used in practice. For example:

<foo href="http://www.example.com/XForms"/>

Also see XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model>

Element, discloses

<xform xmlns="http://www.w3.org/2001/02/xforms" id="Person">

<model href="Schema-Questionnaire.xform" />

```
<instance href="URL-to-retrieve-defaults" />
```

</xform>

and also,

```
<model

    id = xsd:ID

    href = xsd:uriReference

    >

    <!-- Content: (xsd:schema | simple) -->
```

</model>.

It would have been obvious to one of ordinary skill in the art at the time of the

invention to modify Altova XML Spy's validating process, where validates an XML

document against the rules set forth in its Schema or Document Type Definition (DTD)

or it can validate any XML Schema or DTD against the rules set forth in the

corresponding specification of the schema documentation in HTML form, to include the

discovering solution using the URL in the processing instruction (PI) of XForm 1.0 to

improve the validating process of Altova XML Spy for the predictable result of better

Web forms with richer interactions, for online interaction between a XForms Processor

and a remote entity as taught by XForm 1.0 at section 1.1 the Background.)

Claim 31,

The rejection of claim 30 is fully incorporated; In addition, Altova teaches:

a URL is a URL having a path with a suffix that is selected from the

group consisting of .xsf or .xsn.

(See Altova Page 176, teaching Evaluate Xpath, XML Spy includes an XPath Visualizer

that allows you to define/check XPath expressions in your XML documents and display

the results immediately.

See also Altova page 446, teaching Opening a document template *.sps, where

XML Spy Document Editor enables you to edit XML documents based on templates

created in XSLT Designer. Using the broadest reasonable interpretation, the Examiner

reads the claimed **.xsf or .xsn** as equivalent to *.sps as taught by Altova XML Spy.

Since the Applicant Specification discloses, "a URL or URN that will most likely point to

a path having a suffix that is either '*.xsf' (e.g., a manifest or listing other files) or '*.xsn'

(a file that contains multiple files compressed into one file and that are extractable with

the extract '*.exe' utility) See Applicant disclosure Para 102).

Claim 43, Altova does not teach, but XForm 1.0 teaches:

wherein the discovering a solution using the entity in the PI

comprises an act selected from the group consisting of:

(i) discovering the solution using the URL in the PI;

(ii) discovering the solution using a name associated with the

href attribute; discovering the solution using a name in the PI that is

associated with the href attribute; and

(iv) a combination of the foregoing.

(See XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model> Element,

discloses the namespace prefixes "xform:" and "xsd:" are used to denote

the XForms and XML Schema namespaces respectively. This is by convention only;

any namespace prefix may be used in practice. For example:

<foo href="http://www.example.com/XForms"/>

Also see XForm 1.0 Section 9.2 the <xform> Element, and 9.2.1 the <model>

Element, discloses

```
<xform xmlns="http://www.w3.org/2001/02/xforms" id="Person">
    <model href="Schema-Questionnaire.xform" />
    <instance href="URL-to-retrieve-defaults" />
</xform>
```

and also,

```
<model
    id = xsd:ID
    href = xsd:uriReference
>
    <!-- Content: (xsd:schema | simple) -->
</model>.
```

It would have been obvious to one of ordinary skill in the art at the time of the invention

to modify Altova XML Spy's validating process, where validates an XML document

against the rules set forth in its Schema or Document Type Definition (DTD) or it can

validate any XML Schema or DTD against the rules set forth in the corresponding

specification of the schema documentation in HTML form, to include the discovering a

solution using the entity in the PI comprises an act selected from the group consisting

of: (i) discovering the solution using the URL in the PI; (ii) discovering the solution using

a name associated with the href attribute; discovering the solution using a name in the

PI that is associated with the href attribute; of XForm 1.0 to improve the validating

process of Altova XML Spy for the predictable result of better Web forms with richer

interactions, for online interaction between a XForms Processor and a remote entity as

taught by XForm 1.0 at section 1.1 the Background.)

It is noted that any citations to specific, pages, columns, lines, or figures in the prior art

references and any interpretation of the references should not be considered to be

limiting in any way. A reference is relevant for all it contains and may be relied upon for

all that it would have reasonably suggested to one having ordinary skill in the art. See,

MPEP 2123.

Response to Argument

Applicant's Remarks filed on 6/28/2007 with respect to claim 1-45 have been

considered but are moot in view of the new ground(s) of rejection. This office action is a

Final Rejection, because Applicant's amendment necessitated the new ground(s) of

rejection presented in this Office action (see above rejection for details).

It is noted, as requested by the Applicant's attorney (see the Remarks Page 26).

The Examiner has conducted a telephone interview with Applicant's attorney on

09/13/2007.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of
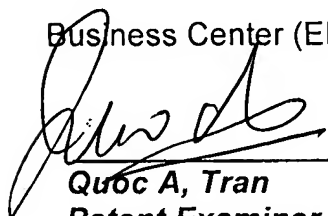
the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Quoc A, Tran
Patent Examiner
Art Unit 2176
09/16/2007

/Doug Hutton/
Supervisory Primary Examiner
Technology Center 2100